Treball de Fi de Grau

**Grau en Enginyeria en Tecnologies Industrials**

# Tecnologia blockchain aplicada a la gestió de cooperatives

**MEMÒRIA**


**Autor:** Arnau Pérez Monné
**Director:** Josep Vilaplana Pastó


**Convocatòria:** 06/2020




Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona


**Resum**

L'objectiu del treball és aprofitar la capacitat de descentralitzar la gestió de les dades que té la tecnologia blockchain per millorar les aplicacions de gestió de cooperatives agràries ja existents. En l'actualitat, tot i que les decisions es prenen de forma assembleària, les aplicacions informàtiques que els hi donen suport tenen una lògica completament centralista. Tant la lectura com el registre de les dades només és accessible per a l'administració de la cooperativa que en té el control absolut.

Els blockchain's son unes bases de dades descentralitzades immutables que mitjançant el consens entre els usuaris generen confiança. Això permet fer transaccions de moneda i/o informació entre els usuaris de manera segura sense necessitat d'un intermediari. Ethereum és un blockchain que permet carregar Smart Contracts, un tipus de programes informàtics que permeten establir unes normes de manera inamovible que la xarxa blockchain s'assegurarà que es compleixin.

Aquest treball consisteix en crear una aplicació que permeti a qualsevol relacionar-se amb un Smart Contract de manera senzilla i la creació d'aquest contracte. La intenció no és crear una aplicació alternativa de gestió sinó una aplicació que pugui treballar paral·lelament amb les que existeixen actualment implementant únicament les funcionalitats que la seva descentralització suposa un avantatge.

El desenvolupament s'ha fet amb la plataforma Embark que ens permet provar el contracte sobre un blockchain simulat. En aquest treball s'arriba a una aplicació funcional i robusta que després d'una ràpida configuració pot ser útil per a la majoria de cooperatives agràries. A través de l'aplicació tant els socis com l'administració de la cooperativa poden seguir el pas de les seves mercaderies per la cooperativa i realitzar algunes operacions a sobre. La sincronització de l'aplicació descentralitzada amb les aplicacions utilitzades actualment no s'ha implementat en aquest treball.

# Sumari

# 1. Glossari

**Blockchain:** Registre distribuït dins el qual estan emmagatzemades cronològicament, en forma de blocs lligats els uns amb els altres, les transaccions successives efectuades entre els seus usuaris, de manera verificable, permanent i anònima.

**Ethereum:** Ethereum és una plataforma de programari de computació distribuïda basada Blockchain i que es distingeix per la funcionalitat de contractes intel·ligents.

**Smart Contract:** Protocol de programari que facilita la negociació de contractes. L'objectiu és proveir de major seguretat i rapidesa, amb menor cost als contractes legals tradicionals. Els contractes intel·ligents estan associats amb el concepte de criptomoneda. La idea inicial va ser proposada per Nick Szabo el 1996.

**Gas:** Unitat de mesura del cost computacional que té realitzar una o varies operacions a Ethereum.

**Ether:** Criptomoneda intrínseca d'Ethereum. Tot els comptes d'Ethereum tenen un balanç d'aquesta moneda i és la utilitzada per pagar el minatge necessari per mantenir la integritat del registre.

**Mining:** Procés de comprovar i afegir transaccions a un blockchain. En el procés diferents miners competeixen per ser el primer de resoldre el problema matemàtic que permet afegir un bloc nou a la cadena. El primer es recompensat amb la moneda intrínseca del blockchain.

**Proof of work:** Sistema utilitzat durant el minatge per evitar comportaments indesitjats, consisteix en requerir una tasca molt pesada computacionalment però molt fàcil de comprovar un cop feta.

# 2. Prefaci

En aquest apartat es pretén presentar el sistema blockchain i el seu funcionament centrant-nos amb Ethereum, el blockchain utilitzat, orientat als lectors no familiaritzats amb aquesta tecnologia per a poder comprendre els apartats següents. A la Bibliografia complementària es poden trobar articles més extensos molt interessants per acabar de comprendre el funcionament d'aquestes bases de dades.

Un blockchain és una base de dades pública que manté un registre de transaccions digitals permanentment. A més, no necessita una autoritat central que en garanteixi la seva immutabilitat, permet la interacció entre usuaris sense la necessitat de confiar en una tercera part.

El blockchain o cadena de blocs s'anomena així per la seva particular manera d'afegir informació a la base de dades. Es comença per un estat inicial sense transaccions desades anomenat gènesis i a partir d'aquest es van afegint transaccions agrupades en blocs. Quan hi ha un cert número de transaccions pendents de registrar-se per sempre al sistema calcula un nou bloc amb aquestes i s'afegeix a la cadena amb informació del bloc immediatament anterior per quedar lligat. D'aquesta manera s'assegura la immutabilitat, no es pot modificar un bloc anterior de la cadena perquè no encaixaria amb els següents.

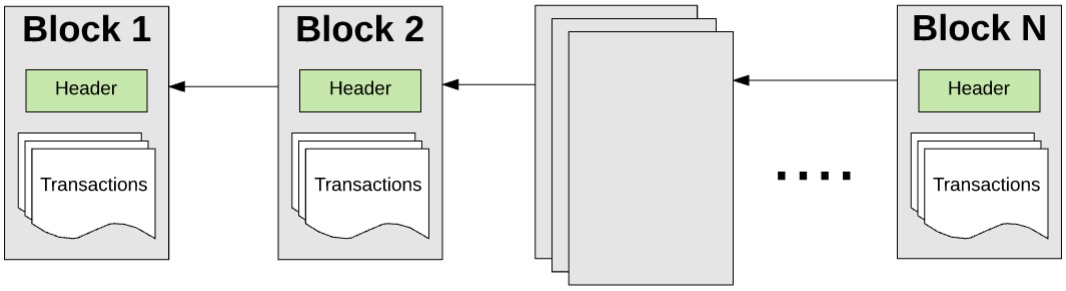



Fig. 1: Cadena de blocs

El procés de calcular un nou bloc no és senzill, és el que s'anomena *mining*, aquesta dificultat és el que assegura que un sol node no pot monopolitzar la creació de nous nodes i d'aquesta manera manipular-los, ja que requeriria d'una capacitat de computació superior a tota la resta de miners junts. En primer lloc cal comprovar que les transaccions que s'ha enviat son vàlides i després també s'han de complir unes condicions addicionals perquè el bloc sencer sigui vàlid. Qualsevol node de la xarxa pot intentar trobar el següent bloc vàlid, constantment hi ha una gran quantitat de nodes mirant de minar el següent bloc.

Per afegir un bloc a la cadena cal ser el primer que aconsegueixes demostrar que tens un bloc vàlid. Ethereum recompensa els miners que tenen èxit amb la seva pròpia moneda

digital anomenada Ether, tots els blockchain tenen la seva pròpia moneda. Cada cop que s'afegeix un bloc nou es crea Ether i s'assigna al miner responsable. La part del minatge més dura computacionalment és calcular la *Proof of Work,* o prova de treball, és un procés llarg i pesat de dur a terme però molt ràpid de comprovar si s'ha fet bé.

Els comptes que poden crear transaccions estan identificats per una adreça de 20 bytes i tenen un estat associat amb la quantitat de moneda que tenen. Hi ha dos tipus de comptes molt diferenciats:

- Comptes controlats externament a través d'una clau privada, els usuaris de la xarxa.
- Comptes contracte, controlats per un codi creats per un compte extern.

Entre usuaris només es poden enviar Ether. En canvi, un usuari pot enviar un missatge a un contracte a més de moneda i que aquest actuï segons està programat, aquestes operacions poden comportar: la transferència de fons, la modificació de variables d'estat, la realització de càlculs, la creació d'un nou contracte… Aquests comptes contracte és la novetat que va introduir Ethereum, els anomenats *Smart Contracts,* comptes

La informació desada en el blockchain és pública, qualsevol node de la xarxa pot tenir tota la informació de la cadena de blocs, però només els miners necessiten tenir-la tota per poder minar. Gràcies a l'estructura de les dades, en forma de Merkel Patricia tree, la resta pot accedir a informació validada sense haver de descarregar tota la base de dades per comprovar-la. El procés de validar part d'un bloc s'anomena "Merkle proof".

La feina del miners la paguen els usuaris en funció de l'ús que en fan. Les operacions tenen associat un cost en Gas segons el cost computacional que tenen, com més complexes son més Gas costen. Aquest unitat existeix per evitar les fluctuacions de valor que té Ether, realment la feina es paga amb Ether, però això és després d'una conversió que s'actualitza constantment. Una mateixa operació sempre costa el mateix Gas, però no el mateix Ether. L'Ether és una unitat molt gran, normalment es traballa amb múltiples del Wei que és 1e-18 Ether.

Un usuari envia juntament amb la transacció el *Gas límit* i el *Gas price*, el producte d'aquests és el màxim valor en Ether que està disposat a gastar-se. Si l'usuari disposa d'aquesta quantitat en el moment d'enviar la transacció aquesta tira endavant amb tot l'Ether indicat. En cas que no s'arribi al *Gas límit* l'usuari recupera el restant, però en cas que es superi el miner es quedarà tot l'Ether enviat, ja que ja ha fet l'esforç de càlcul i revertirà tots els canvis fets fins al moment de l'*Out of Gas*. Si que quedarà registrat que s'ha fet un intent fallit i fins a quin punt ha arribat.

Els miners escullen quines transaccions desen en el següent bloc i les més atractives son les que tenen un Gas price més elevat. Enviar una transacció amb un Gas price inferior al de mercat enrederiria la gravació de la transacció i podria arribar al punt que no es gravés mai.

# 3. Introducció

Les cooperatives agràries son empreses gestionades pels mateixos agricultors que en són socis. Els agricultors posen en comú els seus recursos per realitzar tot tipus d'activitats i operacions encaminades a la millora de la competitivitat de les seves explotacions.

No hi ha un únic model empresarial de cooperativa agrària, en cada cas els socis decideixen quins serveis necessiten i hi ha una infinitat de combinacions possibles. Els diferents serveis es poden agrupar en els que ofereixen[1]:

1. **Crèdit:** Funcionen com a font de finançament per a ampliacions, compra de maquinària, material per a la sembra, etc. Sempre qüestions relacionades amb l'activitat agrícola.
2. **Subministres:** Ofereixen material i maquinària per a la realització de les tasques agràries pròpies.
3. **Ajudes per a la comercialització:** Agrupen les tasques de transformar, empaquetar, distribuir i vendre els seus productes.

Per a la gestió de les activitats anteriors existeix una gran varietat d'aplicacions que permeten adaptar-se a les necessitats.

Tots els programes comparteixen una característica, tenen una estructura centralitzada per guardar les dades. Aquesta organització de la informació porta implícit un problema, tota la informació queda descoberta en cas que la seguretat de la central es vegi compromesa. Els permisos d'edició o lectura de la informació sovint estan centralitzats en una persona o un sistema automàtic, que també es pot veure atacat i pot deixar exposada tota la informació.

El 2009 va sorgir una alternativa a aquesta gestió centralitzada de les dades amb el naixement del ***bitcoin,*** la primera moneda digital viable. Funciona a través del consens entre els usuaris i no necessita cap tercera part que controli tots els moviments i els hi doni validesa. Això és possible gràcies a la tecnologia blockchain, un nou sistema per emmagatzemar i moure informació a través de la xarxa. Una tecnologia amb moltes aplicacions més enllà de les criptomonedes.

El blockchain proposa que cada usuari individualment guardi tota la informació, això no vol dir que tots hi tinguin accés, d'aquesta manera cal comprometre molts sistemes per poder eliminar o canviar qualsevol detall de la informació. A més a més, tots els usuaris tenen clars els permisos que té cadascú, de manera que cap canvi s'accepta si qui el proposa no en té la facultat.

Una de les característiques més interessants del blockchain és la immutabilitat de les dades, un cop una informació es desa, ja mai més es podrà canviar, o com a mínim quedarà constància de tots els canvis. Això és degut a la manera de funcionar, d'on li ve el nom, "cadena de blocs"; una vegada s'afegeix un bloc, queda enllaçat amb els anteriors i

així successivament, de tal manera que el sistema mai permetrà que algú canviï un bloc, ja que no encaixaria amb el següent [2].

Totes aquestes característiques poden ser molt útils en el cas d'entitats que gestionen informació dels seus socis, com ara les cooperatives. Permetria eliminar la necessitat de confiar en una tercera persona que els gestioni les dades, perquè tots serien els propietaris i guardes de tota la informació. Els registres quedarien gravats amb el moment i la persona que els ha realitzat aportant transparència. A més, el fet que la gestió sigui visible per tots els usuaris no és un problema ja que tots en son socis. De fet, la corresponsabilitat forma part de l'empoderament que les cooperatives representen en el sector.

## 3.1. Objectius del projecte

L'objectiu del treball és desenvolupar una aplicació per a gestionar una sèrie d'activitats que es realitzen en una cooperativa agrària. Es vol millorar les aplicacions que hi ha actualment al mercat, introduint la tecnologia blockchain per evitar, en la mesura del possible, la necessitat de confiar amb una tercera persona i aportar transparència a la gestió. Aquest objectiu principal s'abordarà seguint aquests passos:

- Estudi de les aplicacions existents.
- Definició dels requeriments de l'aplicació.
- Elecció de la tecnologia.
- Programació de l'aplicació.
- Test
- Implementació

## 3.2. Abast del projecte

Les aplicacions d'ajuda a la gestió inclouen una gran quantitat de funcionalitats, en la majoria la tecnologia blockchain no té cap sentit, però n'hi ha unes quantes en que un sistema incorruptible podria estalviar molts mal entesos. Aquest projecte es centraria en implementar aquestes funcionalitats de la manera més flexible possible de tal manera que puguin ser útils en el màxim de casos.

Aquestes funcionalitats seria interessant que fossin accessibles des de les aplicacions ja existents com un servei més substituint les funcionalitats escollides, però treballant amb les que no es modifiquen. Aquesta part no s'abordarà en aquest treball, es crearà una aplicació que pugui funcionar paral·lelament a les del mercat.

# 4. Estudi de les aplicacions existents

Existeix una gran varietat d'aplicacions per a la gestió de cooperatives, són aplicacions amb una gran quantitat de funcionalitats. Totes les funcionalitats estan orientades a facilitar la gestió del dia a dia de les cooperatives, optimitzant tasques que serien molt complicades de realitzar sense un suport d'aquest tipus.

Pel que fa compartir les dades, les aplicacions permeten l'extracció de dades en forma de factures per els proveïdors i clients, la generació de models per la administració i fins i tot algunes l'enviament de documentació per correu electrònic.

Algunes permeten la introducció de dades des de varis dispositius autoritzats, d'altres sols permeten des d'un dispositiu central, però en tots els casos la informació queda registrada en un servidor central i no és accessible pels socis.

Totes aquestes aplicacions deixen un gran ventall de funcionalitats potencials de gestionar de forma descentralitzada. A l'**Error! No s'ha trobat l'origen de la referència.** es pot trobar e l llistat de funcionalitats estudiades, indicant en quines aplicacions s'han trobat.

## 4.1. Selecció de funcionalitats

Per seleccionar les funcionalitats més interessants, s'han de tenir clars els avantatges i desavantatges del blockchain.

Segurament la propietat més característica és la immutabilitat de les dades, un cop es desa una informació o dit d'altra manera s'afegeix un bloc a la cadena, aquesta es quedarà així per sempre. De la mateixa manera un cop es carrega un contracte aquest queda immodificable, per això és molt important realitzar una bona bateria de proves abans de pujar-lo. Aquesta nul·la flexibilitat és incompatible amb algunes funcionalitats. Però en canvi li dona resiliència a atacs informàtics permetent guardar informació sensible. Tot i això, no hi ha res més segur que un dispositiu desconnectat de l'exterior, si les dades no s'han de compartir tampoc té cap sentit pujar-les a blockchain, aquesta obvietat ens farà descartar moltes funcions.

Tot i que l'Smart Contract que regula l'entrada i sortida d'informació es carrega compilat, hi ha l'opció de deixar el codi llegible disponible, amb la garantia que és el corresponent, de tal manera que qualsevol pot veure el seu mecanisme intern. Això dona una transparència que genera confiança amb el sistema però pot no interessar en alguns casos. En aquest cas, no hauria de ser un problema que els socis veiessin el funcionament de la seva pròpia cooperativa, en canvi podria no interessar per gestionar la relació amb externs a l'organització.

Per últim, totes les comprovacions que s'han de fer per afegir informació amb garanties

requereixen temps, en el cas del Bitcoin son uns 10 minuts, en canvi a Ethereum son entre 10 i 19 segons. Aquest retard que ni tan sols es previsible també fa descartar funcionalitats que requereixen espontaneïtat.

S'ha escollit la gestió del pas de mercaderies per la cooperativa, l'activitat bàsica d'una cooperativa agrària. El pagès, soci de la cooperativa, aporta el seu producte i la cooperativa el processa en cas de ser necessari i el comercialitza. En l'actualitat, les aplicacions permeten registrar les entrades i sortides, però el feedback cap el soci és en paper. El sistema és segur ja que les dos parts implicades es queden un document conforme s'ha realitzat la transacció, però no gaire pràctic. El que es proposa és informatitzar realment aquesta gestió, eliminant la necessitat d'imprimir i certificar cada entrada físicament mentre es garanteix la seguretat limitant les operacions que es poden fer per part de tots els usuaris i sobretot deixant-ne un registre immutable.

# 5. Definició dels requeriments

L'objectiu d'aquesta aplicació és ser útil per al màxim de cooperatives possibles, la intenció és que sigui fàcilment configurable per adaptar-se a les necessitats particulars de cadascuna.

Totes les accions permeses s'han de poder dur a terme des d'una interfície gràfica còmode que no requereixi coneixements d'informàtica per utilitzar-se. Al tractar-se d'una base de dades immutable convé que abans de registrar qualsevol dada es facin una sèrie de comprovacions per evitar l'error humà en la mesura del possible.

## 3.1 Gestió mercaderies

L'aplicació ha de permetre el seguiment de les mercaderies, de manera que sigui visible per part dels socis en tot moment quina és la situació del seu producte, des de que entra a la cooperativa fins que en surt. Les dades les introduirà una persona autoritzada per la cooperativa i seran visibles per els socis, cada soci podrà veure les seves. Tot això s'ha de desenvolupar amb la tecnologia blockchain per garantir total transparència, el codi quedarà visible i immutable.

### 3.1.1 Entrada

En primer lloc, es registra l'entrada del producte, la informació que és interessant guardar és la següent:

- **Identificació del soci:** Cada soci té associat un número que l'identifica permetent el seguiment de totes les seves transaccions inequívocament.
- **Quantitat:** La quantitat en Kg de producte aportat.
- **Qualitat:** Aquest camp recolliria diferents propietats del producte en funció de quin producte és, per això es deixarà com un camp alfanumèric per permetre la introducció d'un o més valors amb les seves unitats corresponents.
- **Liquidació:** En un principi s'han considerat tres possibles tipus de liquidació del producte, sempre es podria ampliar i/o adaptar a cada cas:
    - **Comercial:** És l'opció més senzilla, es tracta de pactar un preu abans de la venda i liquidar en el moment de l'entrega.
    - **A resultes:** Un cop s'ha acabat la campanya del producte i es saben els costos i ingressos totals, es reparteixen els beneficis entre els socis en funció de la seva aportació (quantitats i qualitats).
    - **Dipòsit:** El soci fa l'entrega i escull el moment de vendre el seu producte, poden ser moments diferents, el preu queda definit per el moment que el soci decideix vendre, ja que és el preu que estableix una administració independent de referència.

- **Producte:** Cal que es puguin escollir totes les varietats de cada producte que accepta la cooperativa.
- **Data d'entrada:** Aquest camp no s'introdueix manualment en el moment de l'entrega, sinó que es guarda automàticament en el moment de registrar-se a la base de dades.

A continuació hi ha un resum dels diferents camps amb les seves propietats requerides:

| Concepte | Tipus de valor | Possibles valors | Validacions | Obligatori |
|---|---|---|---|---|
| **Identificació soci** | Numèric | - | Soci registrat a la BBDD. | Si |
| **Quantitat** | Numèric | - | No superar X dígits. (configurable) | Si |
| **Qualitat** | Alfanumèric | | No superar X caràcters. (configurable) | No |
| **Liquidació** | Opcions | -Comercial<br>-Resultes<br>-Dipòsit | | Si |
| **Producte** | Opcions | Varietats de cada producte | | Si |
| **Data d'entrada** | Data(Automàtic) | - | - | Si |

Taula 1: Dades Nova Entrada

### 3.1.2 Propietat addicional

Un cop feta l'entrega, a vegades s'afegeix informació pel que fa la qualitat del producte, és informació que no es disposa en el moment de l'entrega perquè s'obté d'anàlisis en un laboratori extern. Senzillament es tracta d'un camp amb les mateixes propietats que el de Qualitat de la primera entrada que ha de quedar relacionat amb l'entrada corresponent.

### 3.1.3 Venda

En els casos que s'escull la liquidació en que el soci deix el producte en dipòsit és important que la data escollida pel soci per vendre el seu producte quedi gravada. El programa ha de permetre a un soci escollir el moment de vendre les seves aportacions, per tant ha d'aparèixer un botó a les transaccions pròpies amb aquest tipus de liquidació que no s'hagin venut que guardi la data en que es prem. En cap cas s'ha de permetre la introducció d'una data a escollir, senzillament s'ha de guardar el moment en que es prem el botó.

### 3.1.4 Liquidació

Quan es produeix el pagament de la transacció s'han de guardar una sèrie de camps:

- **Comprador:** És obligatori guardar les dades que permeten trobar l'origen de qualsevol producte alimentari. A més és d'interès tant per l'administrador com per el soci saber on es ven el seu producte.
- **Preu:** És important que quedi registrat a quin preu s'ha venut cada transacció, seria molt interessant que aquest camp es calculés automàticament amb les dades introduïdes a cada transacció, però aquesta funcionalitat supera l'abast d'aquest treball, ja que aquest depèn de molts factors i varien en cada cooperativa. Això ha fet impossible trobar un sistema general que pogués funcionar per a la majoria i per tant es deixaria com una funcionalitat opcional per a incorporar en el moment de configurar l'aplicació a una cooperativa concreta.
- **Data de pagament:** Ha de funcionar com les altres dates, s'ha de guardar el moment en que es registra i no ha de ser modificable per seguretat.

| Concepte | Tipus de valor | Possibles valors | Validacions | Obligatori |
|---|---|---|---|---|
| **Comprador** | Alfanumèric | - | No superar X caràcters. | Si |
| **Preu** | Numèric | - | No superar X dígits. | Si |
| **Data pagament** | Data(Automàtic) | - | - | - |

Taula 2: Dades Liquidació

Un cop liquidada una transacció no es podran realitzar més accions sobre aquesta.

### 3.1.5 Anul·lació

La gràcia d'aquesta aplicació és que un cop entrades les dades ningú pugui modificar-les, però això comporta un problema evident, que passa si s'introdueix una dada incorrecte. Tot i que es faran validacions i es sol·licitarà confirmació abans de gravar qualsevol informació, l'error és possible i s'ha de contemplar.

Per gestionar les transaccions incorrectes s'afegirà un nou estat de la transacció, l'estat anul·lat, les transaccions anul·lades no desapareixeran de la base de dades, però ja no s'hi podran realitzar accions. Els usuaris amb permisos de lectura les podran veure però ja no es podran editar.

## 3.2 Gestió de permisos

Es necessiten dos nivells de permisos, d'ara en endavant els anomenarem administrador i soci. L'administrador ha de poder fer totes les gestions a excepció de decidir el moment de vendre les transaccions en dipòsit, l'única gestió de les que queden registrades que ha de poder fer el soci. L'administrador també té la potestat d'assignar permisos tant afegir com treure socis com canviar l'administrador.

El sistema Ethereum ja té el seu mecanisme de donar d'alta usuaris, validar les dades d'inici de sessió i en cas de ser necessari recuperar contrasenyes perdudes. El que pot fer l'administrador és donar permisos a través de l'adreça pública de cada usuari d'Ethereum sobre l'edició i lectura de dades de la pròpia cooperativa.

En cas que es perdés l'accés a algun usuari cal que hi hagi una solució, per tant es sol·licita que es pugui assignar un identificador de soci a una altre compte d'Ethereum per no perdre els permisos sobre les transaccions ja emeses amb aquest id. També convé que hi hagi una sortida en cas que es perdi l'accés al seu compte d'administrador que suposaria la pèrdua del control de la base de dades, un usuari de recuperació capaç de recuperar els permisos d'administrador en cas de necessitat.

## 3.3 Lectura de dades

A part de les operacions bàsiques que permeten desar informació a la base de dades, hi ha una sèrie d'operacions necessàries per facilitar l'accés a aquesta informació, cal un buscador que permeti localitzar les transaccions emeses.

Per una banda una pantalla que permeti trobar una transacció concreta a partir del seu numero d'identificació i realitzar les accions sobre aquesta que el nivell de permisos de l'usuari li permeti.

Per altre banda, un buscador que mitjançant filtres ens faciliti la llista de transaccions que

compleixen les condicions especificades. Cal fer una diferenciació pel que fa el buscador dels socis i el de l'administrador, l'administrador té accés a totes les transaccions emeses en canvi els socis només les seves pròpies. Els criteris que es consideren convenients per buscar les transaccions son:

- **Data Entrada:** Indicant la un interval de temps mitjançant dues dates, s'han de retornar totes les transaccions que han entrat en aquest període. En cas d'indicar només la primer hauria de retorna les transaccions posteriors a aquesta i en cas d'indicar només la segona mostrar les anteriors.
- **Soci:** Només per el buscador de l'administrador, indicant el número de soci ha de mostrar només les transaccions d'aquest.
- **Comprador:** Indicant el nom del comprador s'han de mostrar totes les transaccions liquidades que s'han venut a aquest comprador.
- **Tipus de liquidació:** S'ha de poder seleccionar un dels tres tipus de liquidació contemplats i mostrar només les transaccions que s'hagin registrat d'aquest tipus.
- **Liquidat:** Mitjançant un check seleccionar només les transaccions que han estat liquidades o les que no han estat liquidades encara.
- **Producte:** Un desplegable que permeti seleccionar les varietats contemplades en l'entrada per mostrar només les transaccions corresponents a la varietat seleccionada.
- **Venut:** Només per a les transaccions en dipòsit, per a diferenciar les transaccions que ja tenen el preu definit i les que no.
- **Anul·lat:** Per defecte només es mostraran les transaccions no anul·lades, però s'ha de permetre veure-les totes o només les anul·lades.

| Criteri | Tipus | Possibles valors |
|---|---|---|
| **Data d'entrada** | Dos camps en format de data | - |
| **Soci** | Numèric | - |
| **Comprador** | Alfanumèric | - |
| **Tipus de liquidació** | Opcions | • Comercial<br>• Dipòsit<br>• Resultes<br>• Totes |
| **Liquidat** | Opcions | • Liquidades<br>• No liquidades<br>• Totes |
| **Producte** | Desplegable | Un producte concret o tots |
| **Venut** | Opcions | • Venudes<br>• No venudes<br>• Totes |
| **Anul·lat** | Opcions | • No anul·lades<br>• Anul·lades<br>• Totes |

Taula 3: Opcions Filtre

Qualsevol combinació de criteris actius a la vegada està permès.

A més a més, d'aquestes pantalles convé tenir una pantalla que contingui filtres molt habituals a escollir per a cada cooperativa. Aquests filtres poden ser diferents per als socis i per a l'administrador, en un principi es tenen aquests:

- Per defecte els administradors tindran un filtre que els mostri les transaccions en dipòsit que el soci ja ha decidit vendre i encara no s'han liquidat, ja que convé que l'administrador estigui al cas d'aquestes transaccions.
- El socis tindrà un filtre predeterminat on es mostraran les seves transaccions en dipòsit que encara no ha venut de manera que les tingui totes presents.

## 3.4 Extracció dades

Tal com ja s'ha comentat l'objectiu d'aquesta aplicació és poder treballar conjuntament amb les aplicacions ja existents al mercat que contenen moltes altres funcionalitats que no té cap interès guardar-les en blockchain. Aquesta comunicació s'hauria d'adaptar a cada aplicació i aquesta personalització del programa no entra a l'abast d'aquest treball, tot i així convé que hi hagi el primer pas desenvolupat, l'extracció massiva de dades a través d'un fitxer.

Aquesta extracció hauria de permetre al usuari poder analitzar el resultat d'una consulta des d'un editor preparat per gestionar grans quantitats de dades.

# 6. Elecció de la tecnologia

En primer lloc, la utilització de la tecnologia blockchain és el que dona sentit al projecte, l'aplicació no farà res de nou, però gràcies a aquesta tecnologia ho farà amb unes garanties que altres tecnologies no poden oferir. Totes les eines al voltant d'aquesta tecnologia acostumen a ser open source, ja que la transparència és la raó de ser de la majoria de projectes i d'aquest. En aquest treball es descartarà qualsevol eina que no ho sigui.

El desenvolupament d'una base de dades descentralitzada no és una tasca senzilla, els mecanismes per validar les dades antigues i afegir-ne de noves a través del consens de la majoria d'usuaris de la xarxa tenen una gran complexitat. Per facilitar aquesta tasca va sorgir Ethereum i l'han seguit d'altres plataformes que permet programar-hi a sobre, afegint el teu contracte, o programa, i que la plataforma distribuïda s'encarregui de la comprovació de les transaccions.

S'utilitzarà Ethereum per la gran comunitat que té a redera, al ser la primera en oferir la possibilitat de programar-hi a sobre és la que disposa de més documentació. En prestacions no és la millor, per exemple, la velocitat de validació és baixa degut al mecanisme de validació que utilitza, però en el cas que ens ocupa tampoc és una prioritat. A més, com que es tracta de generar confiança, el fet que porti 7 anys en funcionament sense que s'hagi detectat cap fissura al sistema és important. Cal dir que el valor de la moneda que distribueix ara mateix és de 26 mil milions de dòlars, per tant de motius per intentar corrompre el sistema no en falten. És el blockchain programable amb més capitalització al mercat, després del Bitcoin que no és programable.

El llenguatge més popular utilitzat a Ethereum és Solidity, es poden programar contractes amb altres llenguatges no específics per aquesta finalitat com C++, JavaScript o Java, però Solidity és el més provat i es va dissenyar específicament per programar Smart Contracts per Ethereum i ara la majoria de blockchain's permeten Solidity. Tot això ens dona seguretat i per això s'ha escollit aquest llenguatge.

Un cop escollida la plataforma en que es pujarà l'Smart Contract, cal escollir l'entorn de programació. Des de Ethereum proposen una llarga llista de recursos disponibles per a desenvolupadors. Hi ha una sèrie d'entorns que només permeten el desenvolupament de l'Smart Contract, com que ens interessa crear un interfície gràfica els descartaríem. A partir d'aquí vaig provar els tres més populars, Ethereum Studio, Truffle i Embark.

Ethereum Studio permet crear una Dapp des del navegador, tot i ser molt atractiva pel fet que al cap d'uns instants de conèixer l'opció ja estàs treballant sobre una aplicació funcional vaig decidir descartar-la per la incompatibilitat amb eines externes. Per a projectes petits probablement seria la millor opció, ja que disposes de totes les eines imprescindibles i de la forma més intuïtiva possible.

Entre Embark i Truffle em va costar més escollir, en prestacions son molt semblants. Totes dues tenen una gran quantitat de documentació i una comunitat molt activa. Em vaig acabar decantant per Embark per dos motius, en primer lloc, tot i que Truffle ara és més popular, Embark va ser la primera i encara té més documentació. En segon lloc, Embark és més configurable, no té una interfície tant amigable com Truffle, però deix més llibertat per configurar l'entorn de proves.

# 7. Programació de l'aplicació

El desenvolupament de la solució té dos parts molt diferenciades, l'Smart Contract i l'aplicació. El Contracte és l'esquelet de l'aplicació, per si sol ja podria realitzar totes les funcions a través d'una consola, però no seria pràctic. L'aplicació té l'objectiu de facilitar l'accés a totes les funcionalitats a través d'una interfície gràfica.

El Contracte és un programa amb variables d'estat que son les que poden variar cada cop que s'afegeix un bloc a la cadena, en cada bloc queda una imatge immutable de l'estat del contracte en aquell moment. Per modificar aquestes variables el contracte té definides una sèrie de funcions d'escriptura amb unes condicions d'execució que si no es compleixen els propis nodes de la xarxa no permetran el canvi. També té funcions de lectura amb les seves pròpies condicions.

Com es veu a la figura de sota, independentment del nivell de permisos que es tinguin, es pot accedir a l'Smart Contract des de l'aplicació o des d'una consola com la que ens proporciona Embark. Qualsevol usuari d'Ethereum pot executar les funcions definides al contracte, però si no compleixen les condicions especificades l'acció no es completarà.

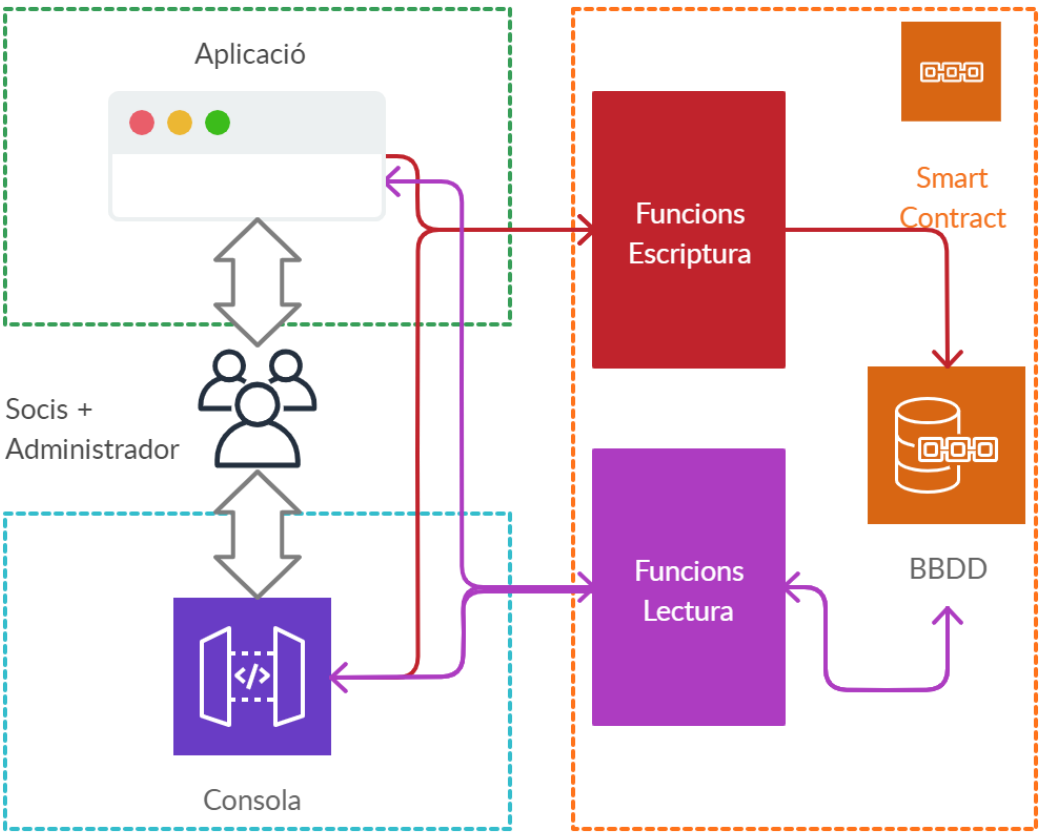



Fig. 2: Esquema implementació

A la figura també es pot veure com les funcions de lectura, les línies liles, tenen fletxa a ambdós costats perquè retornen la informació sol·licitada en cas de complir les condicions. En canvi, les funcions d'escriptura, en vermell, son unidireccionals, s'envia la informació que es vol desar i si els usuaris de la xarxa aproven l'operació es realitzaran els canvis, però com tot això requereix temps en el moment de la sol·licitud no s'obté resposta.

## 7.1. Smart Contract

El primer pas és crear l'Smart Contract per definir la relació que tindran els usuaris amb la base de dades descentralitzada, aquí es defineixen les normes d'edició i de lectura de les dades.

En primer lloc es defineixen les variables d'estat on es guardarà tota la informació de manera immutable, les variables que es guardaran de forma descentralitzada. En el cas que ens ocupa necessitem guardar les transaccions i els usuaris que tenen permisos especials, els socis i l'administrador.

A part de les variables habituals dels llenguatges de programació, Solidity permet crear tipus de variables noves, son simplement agrupacions de les comunes. Aquí es farà servir per crear un nou tipus anomenat *transaccio*. Aquesta nova variable ens permet guardar les diferents propietats que es guarden d'una entrada conjuntament. També s'utilitzarà una estructura nova anomenada *venda* que ens permetrà desar les propietats relacionades amb la liquidació.

Les dades s'organitzen al voltant de la llista de transaccions que conté la informació que es dona d'una en el moment de la seva creació i que ens permetrà recórrer totes les transaccions quan sigui necessari. A partir d'aquí hi ha tres diccionaris diferents, el primer per afegir propietats a posteriori, el segon per guardar la data de cotització en cas de ser necessari i l'últim per liquidar la transacció, els tres diccionaris utilitzen la posició en la llista principal com a clau i s'hi guarda la nova informació. Evidentment no totes les transaccions acabaran tenint una entrada a tots els diccionaris.

Dins de la variable transacció hi ha un camp anomenat estat on s'indica amb una lletra quines dades es tenen d'aquella entrada inicial o dit d'altre manera, a quins diccionaris es pot trobar més informació sobre la transacció. Totes comencen amb una 'e' d'entrada, llavors si pot afegir una 'p' de propietat, una 'c' de data de cotització, una 'l' de liquidat i/o

una “a” de transacció anul·lada.

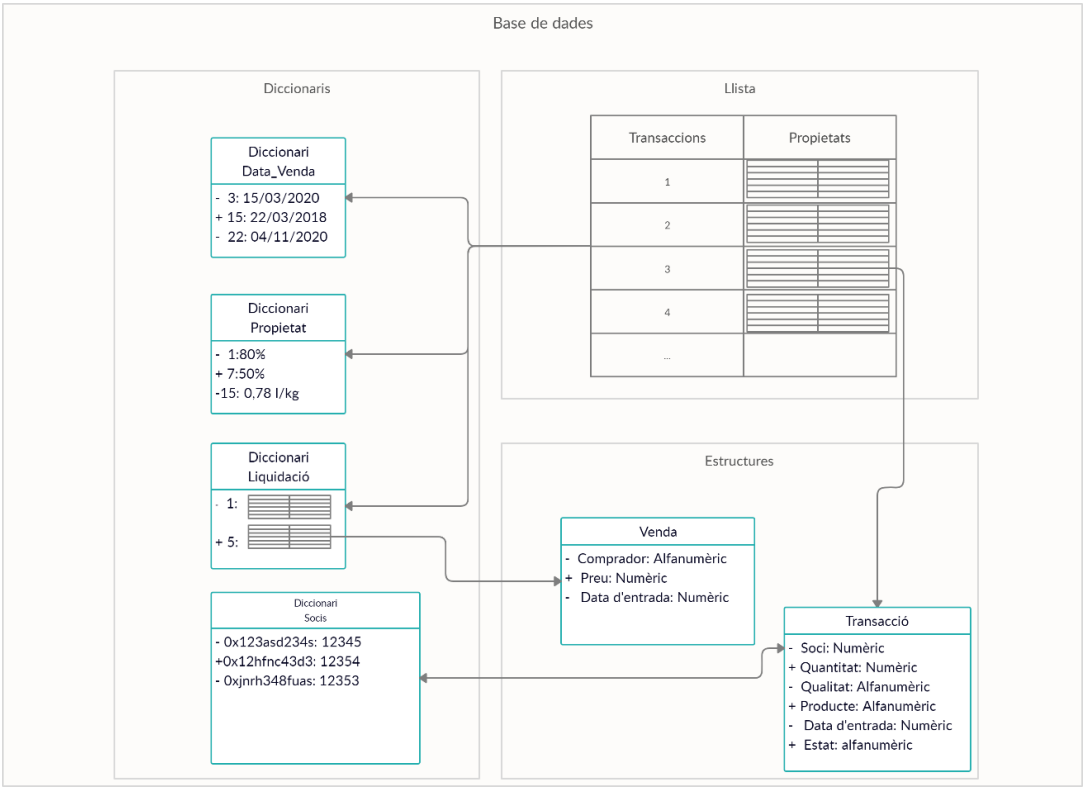



Fig. 3: Esquema BBDD

Aquest sistema que pot semblar més complicat del necessari s’ha dissenyat així per les singularitats de Solidity i per limitar el cost de les operacions.

En Solidity quan es defineix una variable s’ha d’indicar quin tipus és, a més a més crida l’atenció veure com s’indica amb un número el número de bits que s’hi guardarà, si no s’indica per defecte son 256. Però es important no sobredimensionar les variables, ja que no hem d’oblidar que aquestes dades les processaran molts nodes de la xarxa i s’ha de pagar aquest servei.

Tampoc permet concatenar llistes o strings. S’ha de tenir en compte tot això per entendre el contracte. Una altra curiositat del llenguatge és que les funcions no accepten llistes de longitud indefinida ni com a arguments ni com a outputs.

Per optimitzar l’espai les variables alfanumèriques es guardaran en format hexadecimal, en el tipus de variable bytes. Aquest format exigeix que el número tingui el prefix “0x”. Perquè l’usuari no noti aquesta transformació es traduiran les dades a nivell d’aplicació un cop introduïdes o just abans de mostrar-les amb la llibreria web3.

Les dates es guardaran com un enter igual als segons des de l'1 de gener de 1970 fins al moment que es vol desar, aquest format també es convertirà al habitual dd/mm/aaa hh:mm per a la comprensió de l'usuari a l'aplicació.

Un cop creades les variables d'estat hi ha les funcions que ens permeten modificar-les i unes característiques del llenguatge, els *modifiers*. Aquestes últimes es poden afegir a la definició d'altres funcions per fer comprovacions abans d'executar la funció mare, es interessant fer aquestes comprovacions el més aviat possible perquè en cas de no passar la comprovació s'atura l'execució i només es paga per la feina que s'ha fet fins aquell moment.

En aquest contracte hi ha *modifiers* per comprovar que qui crida les funcions té els permisos adequats, hi ha accions que només pot fer l'administrador, d'altres que només el soci sobre les seves transaccions i algunes tan el soci corresponent com l'administrador, però no un usuari qualsevol de la xarxa. També hi ha un *modifier* per comprovar des del principi si la transacció existeix i un altre per comprovar si l'acció que s'està a punt de fer ja ha estat feta o si la transacció ja ha estat liquidada o anul·lada que en bloqueja la resta d'operacions.

Finalment les funcions principals, les d'escriptura i les de lectura, les segones no tenen cost d'execució ja que no requereix capacitat de càlcul. Totes tenen associades alguns *modifiers* per fer les comprovacions més habituals i algunes encara tenen algunes comprovacions extra indiciades amb les funcions *require*(condició,"missatge d'errror") si es vol fer una comprovació o *revert*("missatge d'error") si ja s'ha arribat a un punt que cal tallar la execució. L'interessant d'aquestes funcions és que retornen el motiu de l'error a la consola, però aquesta informació no és accessible des de l'aplicació.

L'administrador per una banda pot afegir socis i canviar d'administrador i per altra pot crear transaccions, afegir-hi propietats, liquidar-les o anul·lar-les. Un soci només pot escollir el moment de vendre una transacció seva en cas que el tipus de liquidació sigui a dipòsit.

Pel que fa les de lectura l'administrador pot llegir qualsevol transacció i el socis només poden veure les seves.

L'operació de canvi d'administrador és delicada, ja que un cop canviat es perd tot el control sobre el contracte i per extensió sobre la base de dades que conté. Tot i que un canvi en un caràcter de l'adreça portaria a una adreça no vàlida i per tant no es registraria el canvi es podria donar el cas de posar una adreça vàlida no desitjada. Per recuperar el control s'ha creat una funció només accessible des de la consola i que només pot executar un usuari indicat abans de carregar el contracte, aquesta funció es pot cridar en qualsevol moment i posa com a administrador el mateix usuari que la crida, l'usuari de recuperació.

La recerca de les transaccions que compleixen els filtres es fa a nivell d'Smart Contract. Com que Solidity sempre ens ha de retornar la mateixa quantitat d'informació, sempre ens retornarà la informació d'una transacció completa. L'usuari definirà un filtre a través de l'aplicació, un cop prem buscar ell només veurà els resultats de la seva cerca a mesura que vagin sortint, pràcticament instantàniament. Per obtenir les transaccions apropiades l'aplicació enviarà el filtre i l'índex de transacció des d'on ha de començar a buscar la base de dades, en un principi l'índex igual a 0, la plataforma blockchain buscarà una transacció que compleixi amb els requeriments del filtre i en cas d'èxit retornarà la informació de la transacció junt amb l'índex de la mateixa. Amb aquesta informació l'aplicació mostrarà la transacció obtinguda a l'usuari i tornarà a sol·licitar una altra transacció a la base de dades, aquesta vegada però l'índex serà el de l'última transacció obtinguda, de manera que no retorni la que ja té. Aquest bucle seguirà fins que ja no obtinguem resposta de la base de dades.

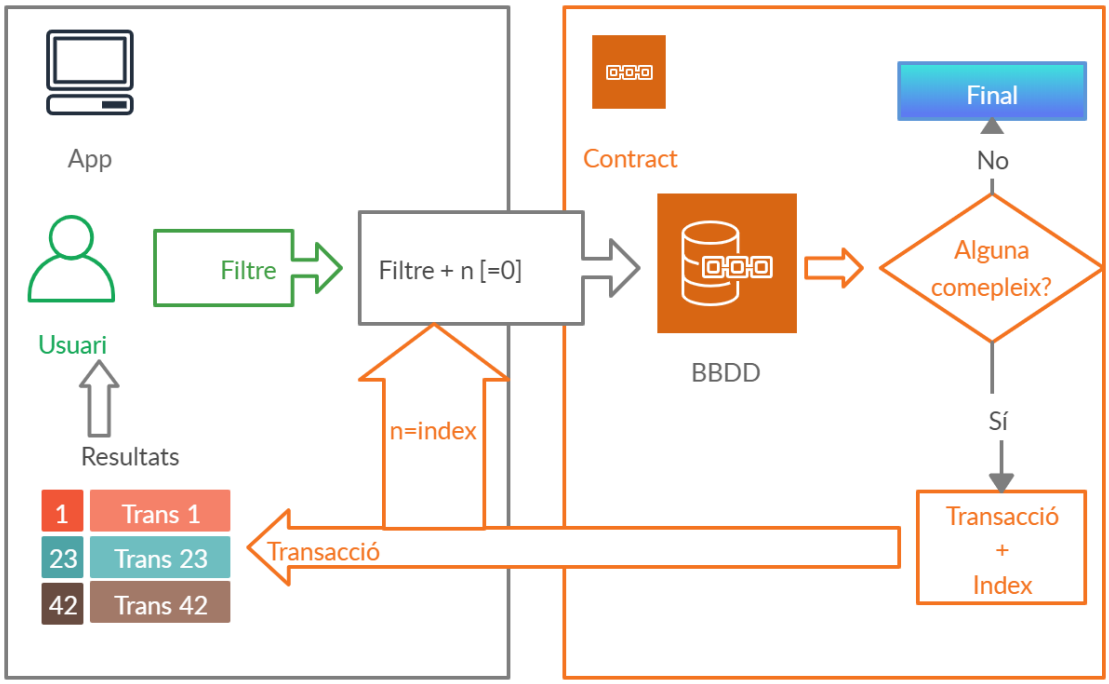



Fig. 4: Algoritme filtre

## 7.2. Aplicació

L'aplicació té com objectiu facilitar el registre i la lectura de dades de la base de dades descentralitzada, bàsicament s'encarregarà de cridar les funcions definides en l'apartat anterior i mostrar els resultats de la manera més entenedora per l'usuari que no té perquè tenir coneixements d'informàtica.

Està desenvolupada amb JavaScript, HTML i CSS, en el desenvolupament de l'aplicació s'han utilitzat les llibrerires Reactstrap i web3. La primera és una llibreria de components de ReactJS, que ens facilita la creació d'una interfície senzilla més ràpidament que les llibreries de JavaScript. Web3 és un conjunt de llibreries que ens faciliten la interacció amb el node d'Ethereum.

En un inici el primer que es fa es comprovar que l'usuari estigui connectat correctament a Ethereum, si no es detecta cap node actiu no es carregarà la pàgina i sortirà el missatge següent:


Fig. 5: Pantalla d'error de connexió

Aquesta comprovació es farà contínuament per evitar errors. En cas que de detectar una connexió activa, el primer que es farà és buscar el nivell de permisos que té l'usuari i en funció de la resposta carregarà unes pestanyes o unes altres, les pestanyes que es carregarien en cada cas son:

| Pestanya | Administrador | Soci |
|---|---|---|
| Pendents | SÍ | SÍ |
| Gestió Permisos | SÍ | No |
| Gestions | SÍ | SÍ |
| Nova Entrada | SÍ | No |

Taula 4: Pestanyes segons permisos

| | | |
|---|---|---|
| **Consultes** | SÍ | SÍ |

Arribats a aquest punt ja es pot entendre l'estructura dels fitxers que componen l'aplicació, es té el fitxer principal anomenat app.js que realitza les funcions descrites fins aquest punt, realitza les comprovacions tan de connexió com de permisos i mostra les pestanyes corresponents. Llavors cada pestanya té associat mòdul guardat a la carpeta de Components que el fitxer principal executa quan es selecciona una pestanya i ens retorna el seu contingut. Aquests mòduls realitzen les seves funcions independentment, a continuació es comenten una sèrie de qüestions generals i després les funcionalitats específiques de cada mòdul.

Els usuaris que accedeixen al Smart Contract a través de l'aplicació a més a més de les validacions que s'incorporen amb Solidity han de passar més validacions, la majoria de les validacions son per evitar errors humans, mentre les validacions definides al contracte estan més orientades al control de permisos. La primera mesura que s'aplicarà a totes les accions que puguin modificar la base de dades és sol·licitar la confirmació de l'operació abans d'enviar-la a la plataforma blockchain. Aquest és el missatge que es mostrarà en el cas de una nova entrada i per a cada operació es té un missatge personalitzat:

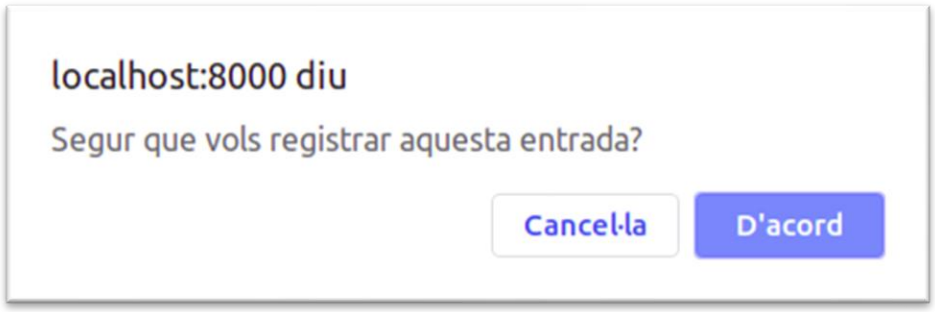



Fig. 6: Missatge confirmació

Totes les comprovacions que ha de fer la xarxa Ethereum abans de gravar una transacció provoquen un retard significatiu entre l'enviament i la confirmació del registre de les dades. Quan s'envia una modificació a la plataforma aquesta no ens retorna una confirmació de l'operació o res semblant, simplement ens retorna un identificador de la transacció, per tant no sabem si s'ha registrat correctament en aquell moment perquè encara s'està processant. En cas d'error tampoc ens retorna el motiu d'aquest.

Per tan cal un mecanisme per confirmar que s'ha tingut èxit, Ethereum té un sistema d'events que ajuda ha visualitzar si s'ha tancat l'operació correctament o no. En aquest treball no s'ha utilitzat aquesta eina ja que degut a la simplicitat del nostre contracte tenim una alternativa, a més a més els events tot i ser un element barat en termes de cost en gas, tenen un cost que no es necessari en aquest cas. El que es farà és a partir del moment en que s'envia una sol·licitud de modificació de la base de dades es començaran a enviar

sol·licituds de lectura buscant les dades que s'acaben d'enviar fins que les trobi registrades, aquestes operacions s'executaran cada 3 segons, recordem que Ethereum grava transaccions cada 10-17 segons.

Des del punt de vista de l'usuari es mostraran uns avisos al peu de la pàgina mentre s'està buscant la confirmació de l'operació i un avis quan l'operació es confirma definitivament, en alguns casos també s'han implementat avisos que informen del motiu de l'error, però al no obtenir el motiu de l'error directament és una tasca difícil i s'ha optat més per intentar evitar-los en la mesura del possible.

Als apartats següents es descriuran les diferents pestanyes.

### 5.3.1 Nova Entrada

En aquesta pestanya, tal com el seu nom indica, es registraran les entrades de productes amb les dades descrites en els requeriments. Les diferents capsules d'entrada de dades ja incorporen el format requerit, de manera que no es poden introduir lletres en el camp ID per exemple.

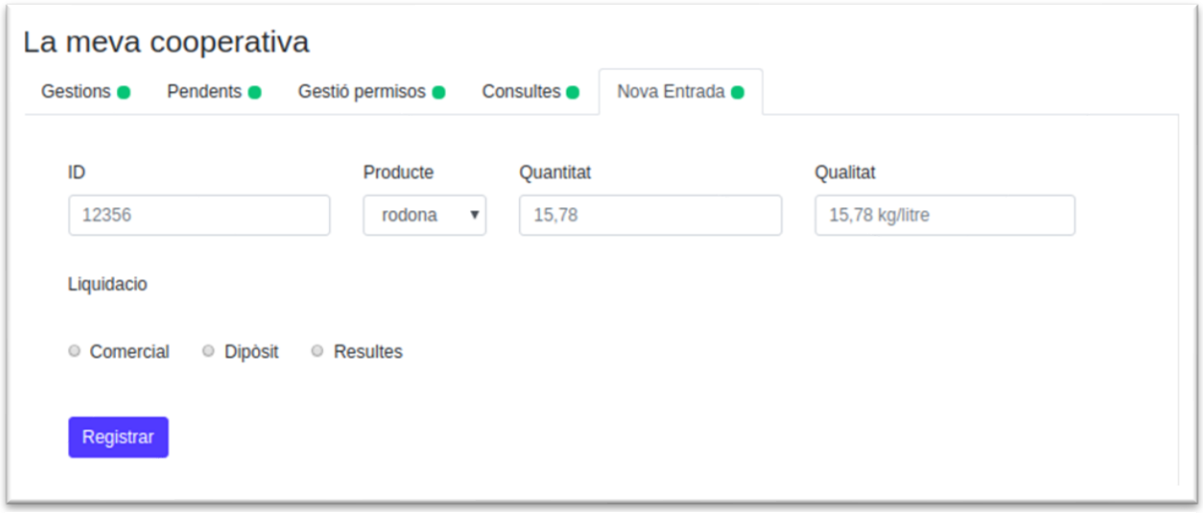



Fig. 7: Pantalla Nova Entrada

No es deixarà registrar una entrada si aquesta no ha informat els camps, ID, Producte, Quantitat i Liquidació, en canvi el camp Qualitat queda opcional. En cas d'intentar desar un registre sense tota la informació es mostrarà el següent avis i no s'enviarà a la

plataforma blockchain:

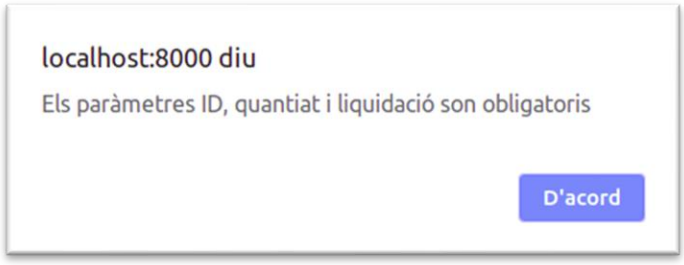



Fig. 8: Missatge paràmetres insuficients Nova Entrada

Un cop s'envia la sol·licitud per registrar una transacció es mostrarà el següent avis al peu de la pàgina:


Fig. 9: Avís registrant entrada

En cas de trobar la transacció registrada correctament es mostrarà el següent avís amb l'índex de la transacció que permetrà identificar-la a partir d'ara:

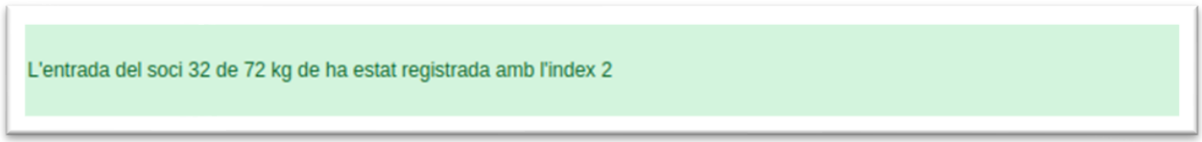



Fig. 10: Avís confirmació entrada

El cas d'aquest avís és peculiar perquè és l'únic que no pot realitzar la comprovació directament. Al no tenir la posició en que es guardarà l'entrada, el que es fa és buscar una transacció registrada més tard de la data en que s'envia que coincideixi amb el número de soci i la quantitat aportada, tot i no ser directa és una comprovació molt segura. Per a les altres comprovacions com que es té a quina transacció van dirigides, es va a buscar

directament la transacció i es comprova si conté la nova informació.

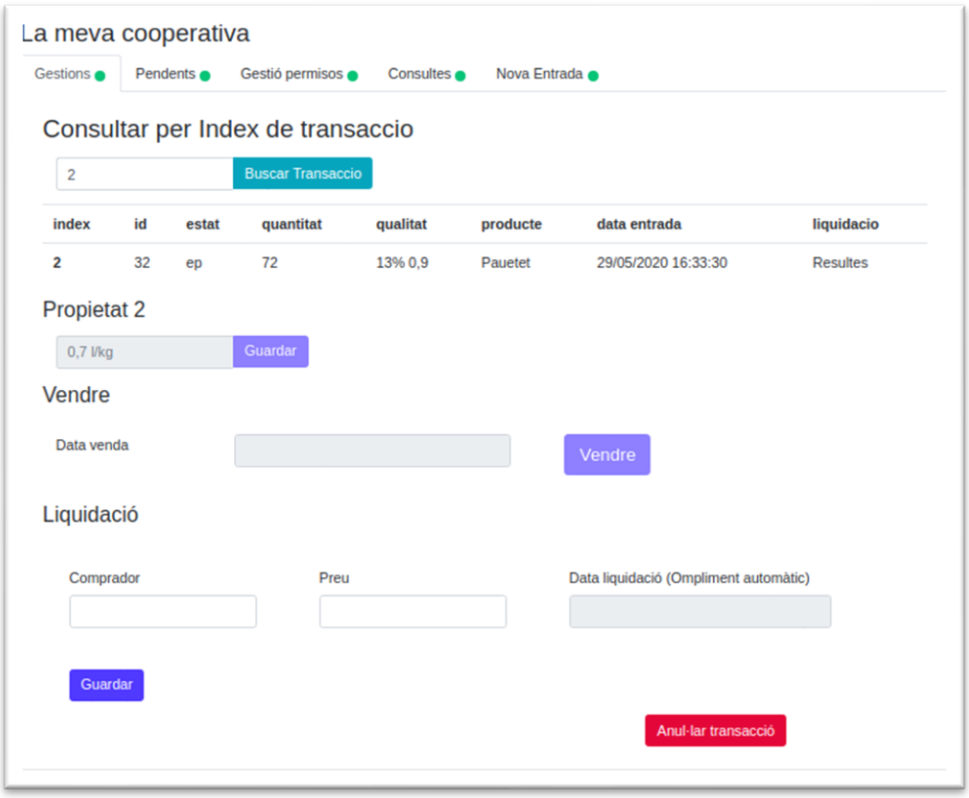
### 5.3.2 Gestions

En aquesta pantalla és poden fer la resta d'operacions que modifiquen la base de dades, els botons disponibles son diferents per a l'usuari administrador i soci. L'administrador pot afegir propietats, liquidar i anul·lar transaccions, en canvi el soci només pot vendre les seves transaccions.

Les operacions només es poden trobar a través del seu índex, assignat per ordre d'entrada, i un cop trobada es carreguen totes les dades disponibles de la mateixa.

Els botons seran visibles per a l'usuari si el seu nivell de permís li permet fer l'operació a alguna transacció, en cas que concretament per la transacció seleccionada aquesta operació no sigui possible es mostraran inhabilitats, això es dona quan l'operació ja ha estat realitzada o quan la transacció ha estat liquidada o anul·lada que queda bloquejada. En cas que l'operació ja s'hagi realitzat, els espais reservats per introduir les dades també estaran bloquejats i contindran les dades guardades a la cadena de blocs. En canvi, les dades desades en el moment de l'entrada del producte es guardaran a la primera línia de dades.

Aquesta pantalla és la única que permet veure totes les dades d'una transacció, les altres pantalles de lectura de dades només mostren les dades que s'introdueixen en crear l'entrada, el que si que mostren totes és l'estat actualitzat de cadascuna de manera que tot i no tenir la informació visible se sap quina existeix i sempre es pot anar en aquesta pantalla per veure-la. A la captura següent es pot veure com es un usuari administrador ja que té tots els botons de les operacions visibles. També es veu com els botons d'afegir propietat i de vendre la transacció estan inhabilitats, el primer perquè l'operació ja s'ha realitzat i podem veure la propietat afegida a la casella. El botó de vendre pot estar inhabilitat per dos motius, en primer lloc la transacció és de liquidació a Resultes, per tant mai tindrà l'operació de Venda disponible, però a més a més per estar habilitat aquest botó l'usuari hauria de ser el propietari de la transacció.




En aquesta pantalla tots els camps són obligatoris per fer les operacions corresponents, menys el camp de la data de liquidació que és d'ompliment automàtic.

Fig. 11: Pantalla Gestions

Pel que fa els avisos aquesta pantalla mostrarà una notificació personalitzada per a cada operació que es faci, des del moment en que s'envia fins que es pot comprovar que s'ha desat correctament es mostraran aquestes notificacions:


Fig. 12: Avisos operacions inacabades

Tots els avisos contenen l'index de la transacció a qui van dirigits i en alguns casos la informació que li estan a punt d'afegir.

A partir que una operació s'ha desat correctament s'esborrarà l'avís anterior i es mostrarà l'avís genèric següent per a totes les operacions indicant l'índex de la transacció:

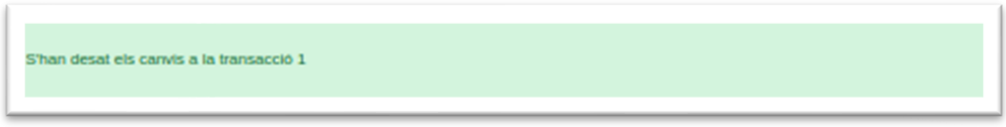



Fig. 13: Avís operació comprovada

Si la transacció seleccionada ja ha estat liquidada o anul·lada es mostraran els següents avisos i tots els botons estaran inhabilitats, amb l'excepció del botó d'anul·lació per a les transaccions liquidades no anul·lades que seguirà habilitat.

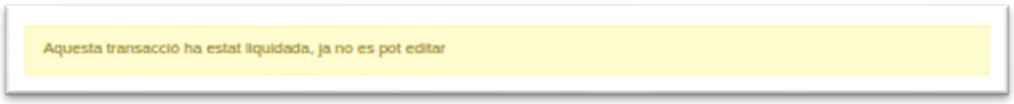



Fig. 14: Avís transacció liquidada


Fig. 15: Avís transacció anul·lada

### 5.3.3 Gestió de permisos

A través d'aquesta pestanya es poden fer dos operacions, canviar l'administrador i associar un usuari d'Ethereum a un id de soci. Evidentment aquestes operacions només estan autoritzades a l'usuari administrador.

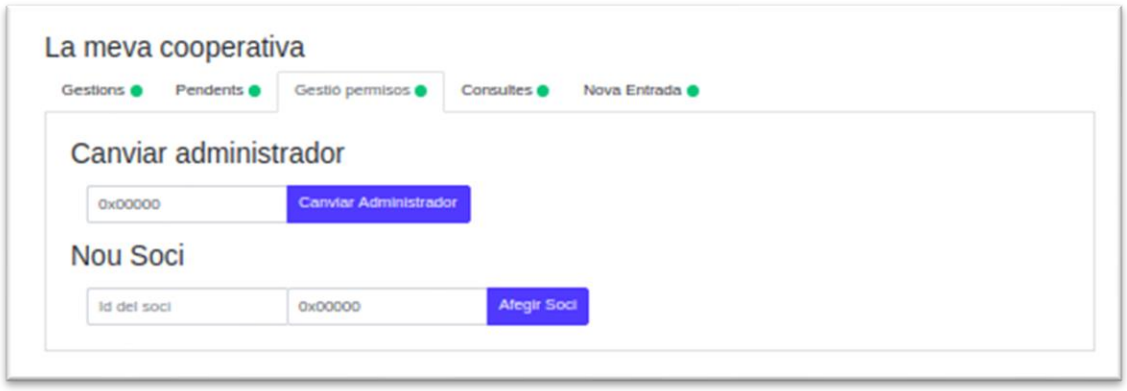



Fig. 16: Pantalla Gestió Permisos

La primera operació és molt senzilla, s'introdueix l'adreça pública d'un usuari d'Ethereum i en prémer el botó s'envia la nova adreça a la base de dades, un cop registrada correctament el primer usuari perdrà els permisos d'administrador, per tant cal anar molt en compte en no equivocar-se, cal dir que si ens equivoquem amb un número o lletra l'adreça no serà vàlida i l'operació no es realitzarà. En cas de indicar una adreça vàlida, però que no en tenim el control perdríem el control del contracte si no fos per la funció de recuperació ja descrita.

Des del punt de vista de l'usuari veurà un avis mentre l'operació no s'ha registrat i un avís quan l'operació conclou:


Fig. 17: Avís Canvi administrador en procés


Fig. 18: Avís Canvi administrador completat

Un cop l'operació s'ha desat l'usuari perd els permisos d'administració i ja no pot realitzar les operacions que podia fer abans. Tot i així, la pàgina no actualitzarà les pestanyes disponibles fins que es recarregui, això no vol dir que durant aquest temps l'usuari mantingui els permisos. Si envia qualsevol operació que ja no tingui permisos aquesta no es registrarà mai i per tant no rebrà confirmació.

L'altre operació disponible és introduir un soci nou, aquesta operació serveix per després poder associar una transacció al seu soci. En aquest punt senzillament es relaciona una adreça d'Ethereum a un id de soci, aquest id de soci és el que la cooperativa coneix i introdueix en el moment de l'entrada d'una transacció, i gràcies a la relació establerta Ethereum serà capaç d'identificar un usuari d'Ethereum com el soci que té permís per veure i/o modificar la transacció.

No es comprova si aquest id ja està registrat, de fet és impossible saber-ho per l'estructura de les dades, però tampoc és un problema. En el cas que un soci perdi l'accés al seu compte de blockchain, totes les seves transaccions quedarien desertes de propietari, en canvi si es pot assignar l'id a una altre adreça el problema queda solucionat.

El que si que no es pot fer, i Ethereum ho comprova, i fins i tot és un error que es mostra a la interfície, és associar dos id a una mateixa adreça, els avisos associats a aquesta operació son:


Fig. 19: Avís soci registrat

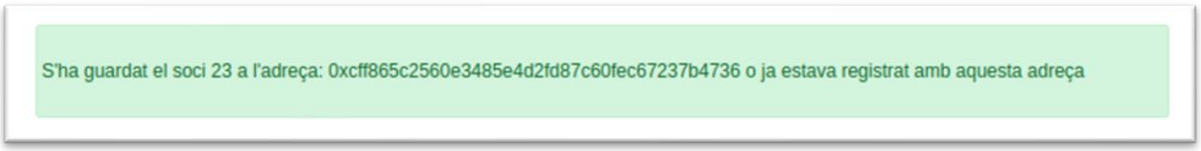



Fig. 20: Avís soci registrat correctament

En cas d'intentar guardar un altre id de soci a una adreça ja registrada:

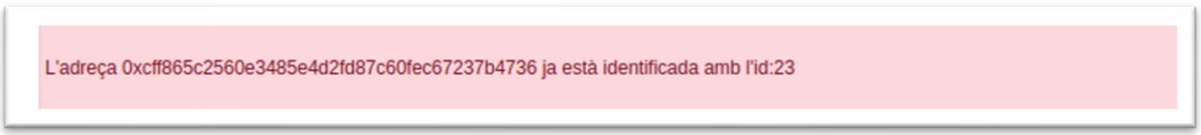



Fig. 21: Avís adreça ja registrada

### 5.3.4 Consultes

En aquest apartat es poden buscar totes les transaccions, sempre que s'hi tingui permís, a través del filtre per facilitar la seva localització.

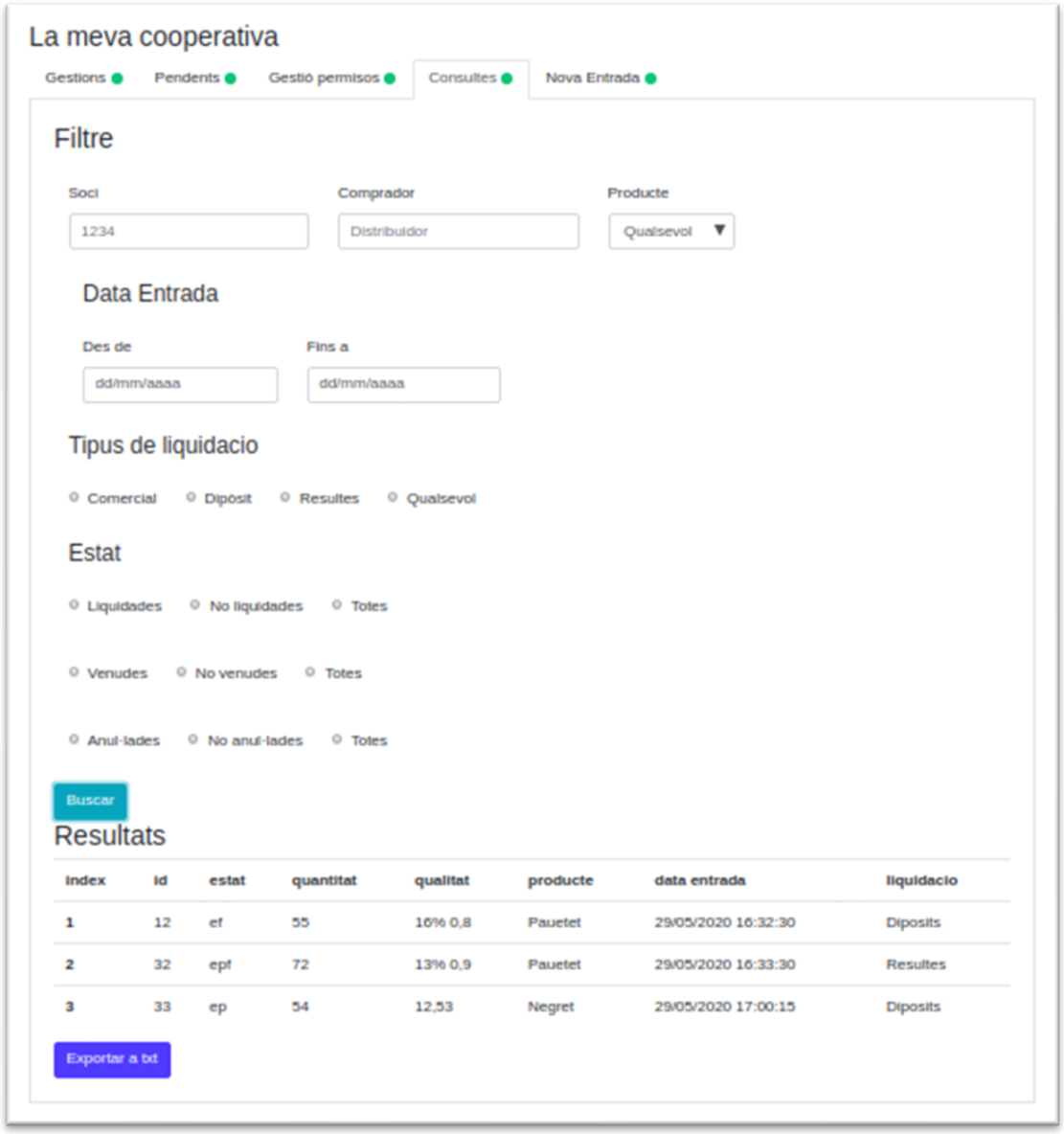



Fig. 22: Pantalla Consultes

Només es mostren les dades introduïdes en el moment de la creació de la transacció perquè sigui homogeni per a totes, però en la casella estat es pot veure si es té més informació de la transacció.

L'única diferència respecta a la pantalla de consultes dels socis és la casella de soci, per als socis automàticament només surten les pròpies i per tant aquesta casella no és necessària.

Un cop obtinguts els resultats és possible descarregar un fitxer txt amb totes les dades obtingudes per poder tractar-lo amb un programa d'anàlisi de dades. El format és molt senzill separa les dades d'una transacció amb una coma i les diferents transaccions amb un punt i coma.

### 5.3.5 Pendents

Aquesta pestanya és un cas particular de la pestanya anterior, on és mostra una cerca amb un filtre preestablert, tal com s'ha dit als requeriments aquest filtre és diferent per als socis i per als administradors. A continuació es mostren les dos pantalles primer la dels administradors i a continuació la dels socis:

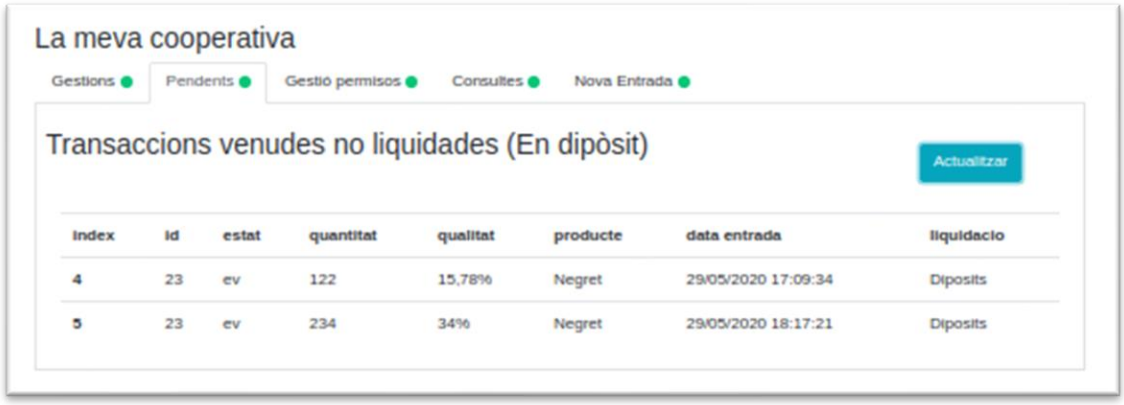



Fig. 23: Pantalla Pendents Administrador

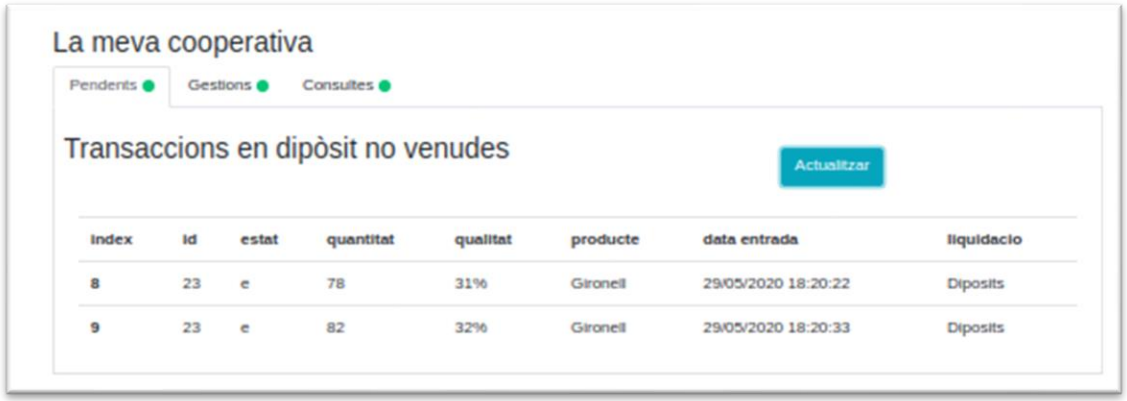



Fig. 24: Pantalla Pendents Socis

# 8. Test

El test, com la programació, té dos parts, la part referent al contracte i la part de l'aplicació, la primera part és imprescindible que no tingui fissures, ja que podria comprometre tota la informació emmagatzemada. En canvi, la segona part és important que funcioni bé, sobretot pel fet que va dirigida a un col·lectiu que no té perquè tenir un gran coneixement d'informàtica i qualsevol error podria portar a malentesos, però si el contracte és robust des de l'aplicació no es podrà fer res que no estigui permès.

Totes les proves s'han fet amb un blockchain local que crea la plataforma Embark per a les proves. L'aplicació s'ha provat des del navegador Chrome tant per a ordinadors com per a mòbils.

A través de les 59 proves descrites a l'Excel adjunt s'ha pogut comprovar un cop finalitzada l'aplicació que el conjunt funciona correctament. Tal com es mostra a l'Excel les proves estan classificades segons diferents criteris, per temàtiques segons el procés i més concretament en operacions, segons el nivell de permisos de l'usuari que ha de realitzar la prova i segons si s'ha de provar a través de l'aplicació o directament accedint al contracte per la consola.

S'han trobat una sèrie d'errors que no son un problema per a la integritat de les dades, però que podrien portar malentesos:

- **Avís de transacció desada correctament:** Quan s'envia una transacció nova si ja existia una transacció amb la mateixa quantitat del mateix soci dona per desada la nova quan troba la vella, la condició de que la data d'entrada sigui posterior a la d'enviament de la transacció no es comprova.
- **Buscador administrador obert pels socis:** A través de la consola es pot accedir al buscador dels administradors sense permís i utilitzar el seu filtre per localitzar qualsevol transacció, es podria solucionar immediatament si no fos que el buscador dels socis depèn del buscador dels administradors i el dels socis deixaria de funcionar si es bloqueges el d'administradors per a la resta d'usuaris.
- **Actualització pestanyes segons permisos:** Un cop hi ha el canvi d'administrador l'administrador vell manté les pestanyes antigues fins que es recarrega la pàgina, està bé que mantingui les pestanyes fins que rep la confirmació que realment s'ha fet el canvi, però convindria que un cop té la confirmació no tingués accés a les pestanyes de les operatives que ja no pot fer perquè els permisos es canvien immediatament.
- **Filtre per data d'entrada:** El filtre on s'indica una data d'entrada mínima, màxima o ambdues no funciona correctament, hi ha un error de format tant per socis com per administradors.

# 9. Implementació

La mateixa plataforma de desenvolupament que s'ha utilitzat per crear l'aplicació i l'Smart Contract en permet gestionar la càrrega a la xarxa Ethereum. Té una interfície gràfica anomenada Embark Cockpit, des d'aquesta plataforma es pot gestionar la càrrega del contracte, totes les operacions que permet l'Smart Contract (sempre que es tinguin permisos), veure totes les transaccions que es graven relacionades amb el contracte i altres funcionalitats molt interessants que complementen el dia a dia que es portaria a través de l'aplicació.

## 9.1. Comptes

Tant per pujar el contracte com per realitzar qualsevol operació però, cal un compte amb els fons suficients per pagar les transaccions. Obtenir un compte d'Ethereum es pot fer des de diferents aplicacions per a mòbil o programes per a ordinador. Ens demanen una contrasenya i una frase de recuperació de 24 paraules, és important escollir una contrasenya segura i desar la frase de recuperació bé, ja que si perdem les dos no hi ha manera de recuperar el control del compte. Amb aquesta informació ens donen l'adreça pública un codi que comença per "0x", conveni per identificar un nombre amb base hexadecimal, seguit de 40 caràcters alfanumèrics del tipus 0x1234FB2EE783A2eE22316b87a16a7274A969319D. Aquesta adreça de 160 bits és la que ens indentifica a la xarxa, permetent-nos enviar Ether a un altre compte o enviar informació al contracte i que aquest sàpiga si tenim permisos. Les mateixes aplicacions ens permeten comprar Ether amb una altre criptomoneda o moneda física.

Hi ha una opció més segura que les aplicacions i programes que impliquen un hardware específic que conté la clau del compte, per les quantitats que es mouen en aquest treball no tindria cap sentit.

## 9.2. Despeses implícites

En aquest apartat s'estudia el cost que tindria desar les dades al blockchain, el cost variable que s'afegiria al dia dia de la cooperativa. El preu que s'ha de pagar als miners per la feina que fan.

Com ja s'ha comentat en el Prefaci, Ether és una criptomoneda que té una valoració molt inestable, per assegurar que la recompensa és sempre atractiva per els miners no es pot fixar un cost en Ether, per això es va crear el Gas una unitat de mesura del cost computacional que té una operació. Un cop es té el cost en Gas es fa un factor de conversió a Ether actualitzat a la cotització de la moneda en aquell moment i és el que es paga al miner.

A la taula següent es mostra un exemple del cost que tindrien unes quantes operacions en la nostra aplicació, s'ha afegit la conversió a l'Euro per tenir una idea del cost real. Els factors de conversió de Gas a Ether i a Euro son els del dia 12/06/2020.

| Operació | Gas | Ether | Euro |
|---|---|---|---|
| **Gravació Contracte** | 2135268 | 0,042705 | 9,01 € |
| **Canvi administrador** | 28405 | 0,000568 | 0,12 € |
| **Recuperació Administrador** | 28741 | 0,000575 | 0,12 € |
| **Afegir Soci** | 44589 | 0,000892 | 0,19 € |
| **Nova Transacció** | 86129 | 0,001723 | 0,36 € |
| **Vendre transacció** | 58913 | 0,001178 | 0,25 € |
| **Afegir propietat** | 59197 | 0,001184 | 0,25 € |
| **Liquidar** | 56987 | 0,00114 | 0,24 € |

Taula 5: Estimació cost operacions

Com es pot veure el cost més important és el de gravar el Contracte, això només s'hauria de fer un cop per cooperativa. Les altres estan al voltant dels 20 ct. d'Euro, destaca la creació d'una nova transacció que se'n va als 36 ct. degut al número elevat de dades que es desen. També és interessant destacar el cost que té l'única operació que poden fer els socis, vendre les transaccions, que té un cost de 25 ct. El cost és petit en comparació amb els diners que es mouen, però s'ha de tenir en compte. Les operacions de lectura no tenen cap cost, ja que no requereixen minatge, això no vol dir que no es facin les validacions indicades en el contracte per executar-les, és fan al node que les sol·licita.

## 9.3. Impacte ambiental

El mecanisme principal de seguretat per garantir la integritat de les dades es basa en fer l'operació d'afegir un bloc nou molt costos computacionalment. El fet que sigui tan pesada l'operació fa que ningú individualment tingui un percentatge significatiu de la capacitat computacional i d'aquesta manera no sempre sigui el primer en obtenir el bloc i el pugui manipular. El cost de fer l'operació tant pesada no és exclusivament econòmic, de fet la principal despesa per els miners és energètica. El consum energètic d'Ethereum sol ja supera països com Zimbabwe o Luxemburg arribant als 7,77 TWh anuals. Es calcula que

el cost energètic d'una transacció sola és 23 kWh quan el consum mitjà diari d'una llar catalana és de 9,31 kWh. Totes aquestes dades s'han extret de la pàgina Digiconomist que a més de monitoritzar el consum de Ethereum també té dades d'altres blockchain's.

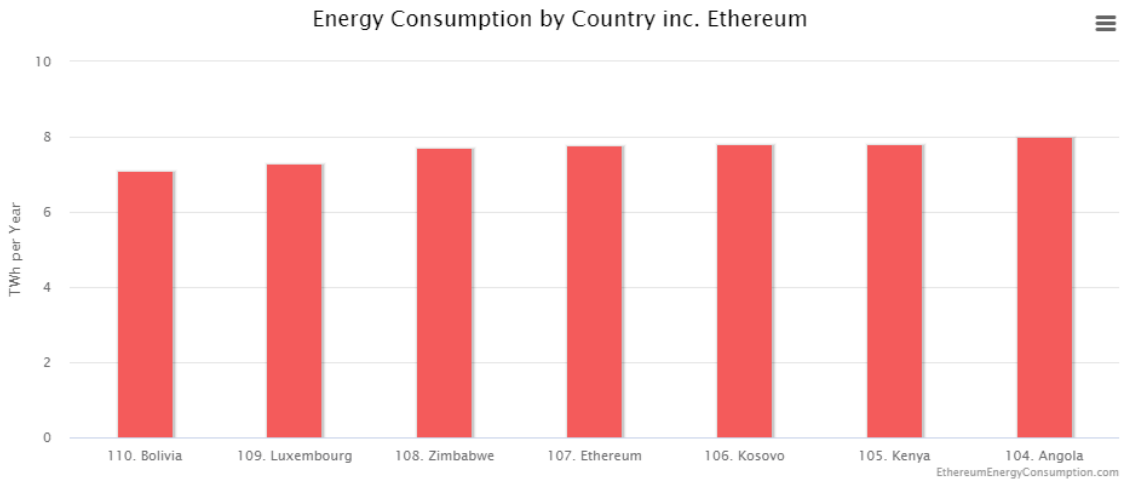



Fig. 25: Comparació consum energètic

Cal tenir en compte això i no fer-ne un ús abusiu, es preveu reduir substancialment el consum energètic optimitzant el procés de minatge sense perjudicar la seguretat, però de moment no és gens menyspreable.

Comparant el cost de les operacions de l'apartat anterior amb el cost econòmic de l'energia consumida per desar-les es pot veure que a un preu de 0,044027 €/kWh (mitjana a Catalunya) una operació mitjana té un cost energètic de 0,41€. Un preu superior al pagat per qualsevol de les transaccions anteriors, això és degut al alt preu de l'electricitat a Catalunya, per aquest motiu les companyies de minatge es concentren a Xina i a Rusia amb un cost de l'energia molt més reduït. Això encara empitjora però l'impacte ambiental de Ethereum, ja que un dels motius de que l'energia sigui més econòmica és que prové de fons més contaminants, molt més barates. Per tant Ethereum consumeix una gran quantitat d'energia elèctrica i a més la competivitat internacional afavoreix que la consumeixi de les fonts més contaminants.

Cal recordar que es tracta d'una tecnologia molt nova que encara té molts reptes per endavant, un dels quals és l'elevat consum energètic, ja s'esta treballant amb la optimització del sistema de minatge per reduir el consum sense comprometre la seguretat.

## 9.4. Privacitat

Una de les principals contrapartides de la transparència que aporta el sistema blockchain és la falta de privacitat de les dades que es tracten, si els miners han de ser capaços de validar les transaccions que es sol·liciten hi han de tenir accés. La privacitat es limita a

l'anonimat, algunes plataformes blockchain han fet avenços en aquesta qüestió, però encara no s'ha aconseguit un sistema realment fiable[4].

Per altre banda, la legislació vigent al respecte, el reglament general de protecció de dades de la Unió Europea(GDPR), estableix drets com el d'eliminació, portabilitat, accés i modificació de dades que no són compatibles amb la característica immutabilitat de les plataformes blockchain[3].

Les dades enviades des de l'aplicació dissenyada en aquest projecte no utilitzen cap tipus de codificació per tant son visibles per tothom, tot i que les funcions per llegir-les estan limitades als permisos descrits durant el treball, això pràcticament només afecta als usuaris de l'aplicació. Qualsevol persona amb certs coneixements sobre blockchain si coneix l'adreça on s'ha desat el contracte pot veure totes les transaccions que s'hi envien amb les corresponents dades que no estan protegides.

En el nostre cas, tot i que no s'envia informació personal que permeti identificar directament una persona al tractar-se d'un col·lectiu petit no seria difícil relacionar la informació present a la base de dades amb un usuari concret, l'anonimització de les dades és molt dèbil.

# 10. Planificació

En aquest apartat es descriuen les etapes seguides en el treball. En primer lloc, al ser una tecnologia completament desconeguda per mi vaig estudiar les seves possibilitats per poder definir els requeriments sabent els seus punts forts i dèbils. Bàsicament va consistir en seguir uns quants cursets sobre la tecnologia i llegir documentació indicada a la Bibliografia complementària .

Un cop sabia com funcionava la tecnologia blockchain vaig anar a informar-me del funcionament d'una cooperativa agrària per seleccionar les funcionalitats més interessants a implementar, aquesta etapa va acabar amb la definició dels requeriments que havia de tenir el sistema, l'apartat 5.

Amb els requeriments definits vaig buscar quina eren els mitjans més adequats per desenvolupat les funcionalitats definides a l'etapa d'Elecció de la tecnologia.

A partir d'aquí ja vaig començar el desenvolupament de l'Smart Contract, el programa que s'ha de carregar a la plataforma blockchain i que regula tant l'estructura de les dades com els permisos d'edició i lectura. Les 80 hores indicades engloben l'aprenentatge del llenguatge de programació escollit, Solidity, i la programació del contracte.

Accedir a la base de dades a través de les funcions definides en el contracte directament des d'una consola no complia els requeriment d'accessibilitat definits en els requeriments. Per tant, vaig desenvolupar una aplicació per cridar aquestes funcions des d'una interfície gràfica senzilla accessible des del navegador tant per dispositius mòbils com ordinadors. Aquesta fase és la que va requerir més temps arribant aproximadament 130 hores.

Un cop tot el desenvolupament fet al Juny, vaig donar el sistema per bo i vaig fer un test general del funcionament per detectar errors difícils de veure durant el desenvolupament concret d'una funcionalitat.

El test el vaig fer amb l'Smart Contract carregat a una plataforma blockchain simulada molt semblant a Ethereum però local. Un sistema que fa les mateixes comprovacions que es fan a Ethereum però només un cop i sense el mecanisme pesat de validació d'un bloc, el minatge. Això permet provar els contractes ràpidament sense requerir una gran capacitat de càlcul, per tant, sense haver de pagar els miners per aquest servei tampoc.

Tot i no arribar a carregar el contracte a la plataforma Ethereum vaig fer un estudi de com seria la implementació real, els passos serien els mateixos que per a la plataforma simulada però requeriria tenir un compte amb els fons suficients. Els costos de la implementació s'han descrit a l'apartat 9.2.

Per últim hi ha la redacció d'aquesta memòria, a la taula de sota es pot veure les diferents etapes amb la dedicació requerida i situades al llarg del quadrimestre.

| Etapa | Hores | Gener | Febrer | Març | Abril | Maig | Juny |
|---|---|---|---|---|---|---|---|
| Estudi tecnologia | 50 | | | | | | |
| Definició requeriments | 20 | | | | | | |
| Programació Contracte | 80 | | | | | | |
| Programació Aplicació | 130 | | | | | | |
| Test | 10 | | | | | | |
| Implementació | 10 | | | | | | |
| Redacció memòria | 40 | | | | | | |
| **Total** | **340** | | | | | | |

Taula 6: Fases treball

A la fase d'implementació faltarien les hores de personalització de l'aplicació per a una cooperativa concreta. Aquesta fase podria ser molt curta, senzillament donant d'alta els usuaris i indicant els productes que accepta la cooperativa, o es podria allargar molt si es sol·licita sincronitzar-se amb una altre aplicació ja existent, noves funcionalitats, modificació d'existents, nous perfils d'usuaris...

# 11. Pressupost

Al tractar-se d'un producte purament de software on només s'ha utilitzat programari lliure per desenvolupar-lo, el cost del projecte és bàsicament el de les hores dedicades.

El temps dedicat s'ha agrupat en dos conceptes, per una banda les hores d'analista on hi ha les d'estudi de la tecnologia, definició dels requeriments i test i per altra les hores de programador amb les hores de programació de l'aplicació i el contracte i la implementació. S'ha considerat el preu hora orientatiu per aquestes professions per als principiants.

El preu d'amortització del material s'ha calculat a partir de l'esperança de 3,5 anys d'un portàtil que és el material utilitzat.

| Concepte | Unitats [h] | Preu/unitat [€/h] | Total |
|---|---|---|---|
| **Hores analista de sistemes informàtics** | 100 | 14,44 | 1444,00€ |
| **Hores programador** | 240 | 13,36 | 3.206,4€ |
| **Amortització material** | 300 | 0,3 | 90€ |
| **Total** | 340 | - | **4.740,4€** |

# 12. Conclusions

L'aplicació malgrat alguns errors funciona correctament, després d'una ràpida configuració podria entrar en funcionament, per sincronitzar-se amb una altre aplicació de gestió requeriria més desenvolupament però és possible. S'ha desenvolupat una aplicació senzilla que pot utilitzar qualsevol tan des de mòbils com des d'un ordinador i robusta pel que fa la immutabilitat de les dades, però amb aspectes a millorar sobretot pel que fa la privacitat de les dades.

Convindria introduir un sistema d'encriptació de les dades per garantir realment els permisos de lectura, actualment només garantits per als usuaris que hi accedeixen per l'aplicació. Aquesta tasca no és senzilla, ja que part del sentit del projecte és poder deixar tot el codi obert a qualsevol que el vulgui mirar i d'aquesta manera assegurar quin és el seu funcionament. I si el contracte ha de tenir accés a les dades qualsevol que en vegi el funcionament també en tindrà. Una solució parcial seria encriptar les dades que el programa no ha d'entendre abans d'enviar-les a aquest, hi ha dades que el contracte no utilitza per res senzillament en garanteix la seva immutabilitat, aquestes es podrien introduir encriptades.

A través de la tecnologia blockchain s'ha aconseguit una base de dades immutable, per molt que es puguin variar dades com qui és l'administrador de la base de dades sempre quedarà registrat qui ho va ser i quan. Pel que fa les transaccions tot i que el registre segueix centralitzat a l'encarregat de la cooperativa tots el moviments que fa queden registrats i estan limitats a les operatives justes i necessàries. Aquesta transparència té un cost per a la cooperativa, ja que perd flexibilitat alhora de treballar, però crec es pot vendre com un servei addicional que no té perquè ser integral, podria funcionar només per els socis que ho sol·licitin. Per als socis que fan aportacions petites o molt puntuals segurament no té sentit. Eliminant la necessitat d'imprimir moltes factures per certificar les transaccions i facilitant l'accés a les dades.

Tota la responsabilitat que tenia la gestió de la cooperativa la segueix tenint, pel que fa la descentralització de la presa de decisions importants no s'ha pogut avançar. Això és degut al plantejament que s'ha seguit des del principi de ser útil per el màxim de cooperatives amb un mínim de configuració. Per les particularitats que té cada cooperativa no s'ha trobat per exemple un mecanisme de càlcul del preu dels productes en base a les dades objectives que es tenen. Segurament l'error ha sigut intentar adaptar el programa al funcionament actual de les cooperatives i no proposar una altre manera de funcionar, fins ara impossible, però que la tecnologia blockchain ara ens permet.

Pel que fa la tecnologia blockchain, he pogut constatar que es tracta d'una tecnologia nova que es troba en desenvolupament amb molt d'espai per recórrer i amb un gran potencial, però amb problemes importants actualment. Sobretot en com donar transparència sense perdre la privacitat dels usuaris i en la gran despesa energètica que suposa.

# Agraïments

En primer lloc, vull agrair a la cooperativa de Castellvell del Camp, concretament al seu gestor Lluís Pallejà, per la seva ajuda i orientació que han estat essencials per comprendre el funcionament d'una cooperativa i les seves necessitats.

També agraeixo l'orientació oferta pel meu tutor del Treball de Final de Grau, Josep Vilaplana, per les seves indicacions i correccions donades durant la realització del treball.

# Bibliografia

## Referències bibliogràfiques

**[1]** FEDERACIÓ DE COOPERATIVES AGRÀRIES DE CATALUNYA. *Què és una cooperativa agrària.* 2012. *Institució. Departament(s). [http://www.cooperativesagraries.cat/ca/que-es-una-coop-agraria.html, 27 de gener de 2020]*

**[2]** ZHENG, Zibin & XIE, Shaoan & DAI, Hong-Ning & CHEN, Xiangping & WANG, Huaimin. *An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends.* 6th IEEE International Congress on Big Data, 2017.

[https://www.researchgate.net/publication/318131748_An_Overview_of_Blockchain_Technology_Architecture_Consensus_and_Future_Trends, 27 de gener de 2020]*

**[3]** Teles, Duarte. Data Protection with Ethereum Blockchain. 2018.

[https://www.researchgate.net/publication/334409306_Privacy_with_Ethereum_smart_contracts,Juliol de 2019]*

**[4]** Khan, Nida & Nassar, Mohamed. A Look into Privacy-Preserving Blockchains. 2019.

[https://www.researchgate.net/publication/335320780_A_Look_into_Privacy-Preserving_Blockchains, Agost de 2019]*

# Bibliografia complementària

Per a conèixer més sobre Ethereum:

Buterin, Vitalik. Ethereum Whitepaper. 2013.

[https://ethereum.org/whitepaper/]*

Wood, Gavin & Savers, Nick. Ethereum Yellow Paper. 2018.

[https://github.com/ethereum/yellowpaper, 2018]*

Kasireddy, Preethi. How does Ethereum work, anyway?. 2017.

[https://medium.com/@preethikasireddy/how-does-ethereum-work-anyway-22d1df506369, Setembre de 2017]*